

---

# Differentially Private Variational Inference for Non-conjugate Models

---

Joonas Jälkö<sup>1</sup>, Onur Dikmen<sup>1</sup> and Antti Honkela<sup>1,2,3</sup>

<sup>1</sup> Helsinki Institute for Information Technology (HIIT), Department of Computer Science

<sup>2</sup> Department of Mathematics and Statistics <sup>3</sup> Department of Public Health

University of Helsinki

## Abstract

As collecting huge amounts of personal data from individuals has been established as a standard nowadays, it is really important to use these data in a conscientious way. For example, when performing inference using these data, one has to make sure individuals' identities or the privacy of the data are not compromised. Differential privacy is a powerful framework that introduces stochasticity into the computation to guarantee that it is difficult to breach the privacy using the output of the computation. Differentially private versions of many important machine learning methods have been proposed, but still there is a long way to pave towards an efficient unified approach applicable to handle many models. In this paper, we propose a differentially private variational inference method with a very wide applicability. The variational inference is based on stochastic gradient ascent and can handle non-conjugate models as well as conjugate ones. Differential privacy is achieved by perturbing the gradients. We explore ways to make the algorithm more efficient through privacy amplification from subsampling and through clipping the gradients to limit the amount of information they leak. We explore the effect of different parameter combinations in logistic regression problems where the method can reach an accuracy close to non-private level under reasonably strong privacy guarantees.

## 1 Introduction

Recent years have seen a boom in collection of digital data due to many technological advances but mostly processing power, automatised collection and easy storage. Machine learning methods running on larger data sets provide better estimates with more generalisation power. With more people getting more tightly involved in the ubiquitous data collection, privacy concerns related to the data are becoming more important. People will be much more willing to contribute their data if they can be sure that the privacy of their data can be protected.

Differential privacy (DP) (Dwork, 2006; Dwork and Roth, 2014) is a strong framework with strict privacy guarantees against attacks from adversaries with arbitrary side information. The main principle is that output of an algorithm (such as a query or an estimator) should not change much if the data for one individual is modified or deleted. This can be accomplished through adding stochasticity at different levels of the estimation process, such as adding noise to data itself (input perturbation), changing the objective function to be optimised or how it is optimised (objective perturbation), releasing the estimates after adding noise (output perturbation) or by sampling from a distribution based on utility or goodness of estimates (exponential mechanism).

A lot of ground breaking work has been done on privacy-preserving versions of standard machine learning approaches, such as objective-perturbation-based logistic regression (Chaudhuri and Monteleoni, 2008), regression using functional mechanism (Zhang et al., 2012) to name a few. However, privacy preserv-

ing Bayesian inference has not accomplished such big leaps. There are important work on incorporating the above mechanisms into probabilistic modelling and handling them in a Bayesian setting, such as (Williams and McSherry, 2010; Zhang et al., 2014). One posterior sample (Wang et al., 2015) and posterior perturbation (Dimitrakakis et al., 2014; Zhang et al., 2016) have been important steps towards private Bayesian learning, however they suffer from lacking asymptotic efficiency, i.e., learning does not optimally benefit from having larger number of data points. Foulds et al. (2016) proposed an asymptotically efficient private Gibbs sampling method based on perturbing sufficient statistics of data. This approach is applicable to models where non-private inference can be performed by accessing sufficient statistics.

In this paper, we introduce a privacy-aware variational Bayesian (VB) inference method. Sufficient statistics perturbation idea of Foulds et al. (2016) can also be applied to models where VB makes use of only those statistics from data. The goal in this work is to tackle the general case. A differentially-private VB method (DPVB) is proposed based on gradient perturbation, which can be seen as a form of objective perturbation. The method does not make any assumptions on conjugacy in models and whether data are accessed individually or not. We make a thorough case study on the Bayesian logistic regression model with comparisons to the non-private VB under different design decisions for DPVB.

The paper is organised as follows: In Section 2 the basics of DP and VB are explained and the proposed method, DPVB, is explained in Section 3. The case study, Bayesian logistic regression model, and DPVB derivation for it are given in Section 4. Detailed experiments are carried out in Section 5 to analyse the effect of different parameter and design settings. Section 6 concludes the paper with direction of future research.

## 2 Background

### 2.1 Differential privacy

Differential privacy (DP) (Dwork, 2006) is a framework that provides mathematical formulation for privacy that enables proving strong privacy guarantees.

**Definition 1** ( $\epsilon$ -Differential privacy). A randomised algorithm  $\mathcal{A}$  is  $\epsilon$ -differentially private if for all pairs of adjacent data sets, i.e., differing only in one data sample,  $x, x'$ , and for all sets  $S \subset \text{im}(\mathcal{A})$

$$\Pr(\mathcal{A}(x) \in S) \leq e^\epsilon \Pr(\mathcal{A}(x') \in S).$$

There are two different variants depending on which data sets are considered adjacent: in *unbounded DP*

data sets  $x, x'$  are adjacent if  $x'$  can be obtained from  $x$  by adding or removing an entry, while in *bounded DP*  $x, x'$  are adjacent if they are of equal size and equal in all but one of their elements (Dwork and Roth, 2014). The definition is symmetric in  $x$  and  $x'$  which means that in practice the probabilities of obtaining a specific output from either algorithm need to be similar. The privacy parameter  $\epsilon$  measures the strength of the guarantee with smaller values corresponding to stronger privacy.

$\epsilon$ -DP defined above, also known as *pure DP*, is sometimes too inflexible and a relaxed version called  $(\epsilon, \delta)$ -DP is often used instead. It is defined as follows:

**Definition 2** ( $(\epsilon, \delta)$ -Differential privacy). A randomised algorithm  $\mathcal{A}$  is  $(\epsilon, \delta)$ -differentially private if for all pairs of adjacent data sets  $x, x'$  and for every  $S \subset \text{im}(\mathcal{A})$

$$\Pr(\mathcal{A}(x) \in S) \leq e^\epsilon \Pr(\mathcal{A}(x') \in S) + \delta.$$

It can be shown that  $(\epsilon, \delta)$ -DP provides a probabilistic  $\epsilon$ -DP guarantee with probability  $1 - \delta$  (Dwork and Roth, 2014).

#### 2.1.1 Composition theorems

One of the very useful features of DP compared to many other privacy formulations is that it provides a very natural way to study the privacy loss incurred by repeated use of the same data set. Using an algorithm on a data set multiple times will weaken our privacy guarantee because of the potential of each application to leak more information. In fact in worst case if our algorithm is  $(\epsilon, \delta)$ -DP, then  $k$ -fold composition of that algorithm provides  $(k\epsilon, k\delta)$ -DP. More generally releasing joint output of  $k$  algorithms  $\mathcal{A}_i$  that are individually  $(\epsilon_i, \delta_i)$ -DP will be  $(\sum_{i=1}^k \epsilon_i, \sum_{i=1}^k \delta_i)$ -DP. Under pure  $\epsilon$ -DP when  $\delta_1 = \dots = \delta_k = 0$  this is the best known composition that yields a pure DP algorithm.

Moving from the pure  $\epsilon$ -DP to general  $(\epsilon, \delta)$ -DP allows a stronger result with a smaller  $\epsilon$  at the expense of having a larger total  $\delta$  on the composition. This trade-off is characterised by the Advanced composition theorem of Dwork and Roth (2014, Theorem 3.20), which becomes very useful when we need to use data multiple times

**Theorem 1** (Advanced composition theorem). *Given algorithm  $\mathcal{A}$  that is  $(\epsilon, \delta)$ -DP and  $\delta' > 0$ ,  $k$ -fold composition of algorithm  $\mathcal{A}$  is  $(\epsilon_{tot}, \delta_{tot})$ -DP with*

$$\epsilon_{tot} = \sqrt{2k \ln(1/\delta')} \epsilon + k\epsilon(e^\epsilon - 1) \quad (1)$$

$$\delta_{tot} = k\delta + \delta'. \quad (2)$$

The theorem states that with small loss in  $\delta_{tot}$  and with small enough  $\epsilon$ , we can provide more strict  $\epsilon_{tot}$

than just summing the  $\epsilon$ . This is obvious by looking at the first order expansion for small  $\epsilon$  of

$$\epsilon_{tot} \approx \sqrt{2k \ln(1/\delta')} \epsilon + k\epsilon^2.$$

In this paper we are using data iteratively over many iterations so the advanced composition becomes necessary.

### 2.1.2 Gaussian mechanism

There are many possibilities how to make algorithm differentially private. In this paper we use *objective perturbation*. We use the *Gaussian mechanism* as our method for perturbation. Theorem 3.22 of Dwork and Roth (2014) states that given query  $f$  with  $\ell_2$ -sensitivity of  $\Delta_2(f)$ , releasing  $f(x) + \eta$ , where  $\eta \sim N(0, \sigma^2)$ , is  $(\epsilon, \delta)$ -DP when

$$\sigma^2 > 2 \ln(1.25/\delta) \Delta_2^2(f) / \epsilon^2. \quad (3)$$

The important  $\ell_2$ -sensitivity of a query is defined as:

**Definition 3** ( $\ell_2$ -sensitivity). Given two adjacent data sets  $x, x'$ ,  $\ell_2$ -sensitivity of query  $f$  is

$$\Delta_2(f) = \sup_{\substack{x, x' \\ \|x - x'\|_2 = 1}} \|f(x) - f(x')\|_2.$$

### 2.1.3 Privacy amplification

We use a stochastic gradient algorithm that uses subsampled data while learning, so we can make use of the amplifying effect of the subsampling on privacy. This *Privacy amplification theorem* (Li et al., 2012) states that if we run  $(\epsilon, \delta)$ -DP algorithm  $\mathcal{A}$  on randomly sampled subset of data with uniform sampling probability  $q > \delta$ , privacy amplification theorem states that the subsampled algorithm is  $(\epsilon_{amp}, \delta_{amp})$ -DP with

$$\epsilon_{amp} = \min(\epsilon, \log(1 + q(e^\epsilon - 1))) \quad (4)$$

$$\delta_{amp} = q\delta, \quad (5)$$

assuming  $\log(1 + q(e^\epsilon - 1)) < \epsilon$ .

## 2.2 Variational Bayes

Variational Bayes (VB) methods (Jordan et al., 1999) provide a way to approximate the posterior distribution of latent variables in a model when the true posterior is intractable. True posterior  $p(\theta|\mathbf{x})$  is approximated with a variational distribution  $q_\xi(\theta)$  that has a simpler form than the posterior, obtained generally by removing some dependencies from the graphical model such as the fully-factorised form  $q_\xi(\theta) = \prod_d q_{\xi_d}(\theta_d)$ .  $\xi$  are the variational parameters and their optimal values  $\xi^*$  is obtained through minimising the Kullback-Leibler (KL) divergence between  $q_\xi(\theta)$  and  $p(\theta|\mathbf{x})$ . This is also equivalent to maximising *evidence lower bound* (ELBO).

**Definition 4** (ELBO). Given joint distribution  $p(\mathbf{x}, \theta) = p(\mathbf{x}|\theta)p(\theta)$ , ELBO of  $q_\xi$  is given as follows

$$\begin{aligned} \mathcal{L}(q_\xi) &= \int q_\xi(\theta) \ln \left( \frac{p(\mathbf{x}, \theta)}{q_\xi(\theta)} \right) \\ &= -\text{KL}(q_\xi(\theta) \| p(\theta)) + \langle \ln p(\mathbf{x}|\theta) \rangle_{q_\xi(\theta)} \\ &= -\text{KL}(q_\xi(\theta) \| p(\theta)) + \sum_i \langle \ln p(x_i|\theta) \rangle_{q_\xi(\theta)}, \end{aligned}$$

where  $\langle \cdot \rangle_{q_\xi(\theta)}$  is expectation taken w.r.t  $q_\xi(\theta)$ .

When the model is in the conjugate exponential family (Ghahramani and Beal, 2001) and  $q_\xi(\theta)$  is factorised, the expectations that constitute  $\mathcal{L}(q_\xi)$  are analytically available and each  $\xi_d$  is updated iteratively by fixed point iterations. Most popular applications of VB fall into this category, because handling of the more general case involves more approximations, such as defining another level of lower bound to ELBO or estimating the expectations using Monte Carlo integration.

In this paper we focus on model that are not conjugate and exact variational updates are thus not tractable. We need to fit the approximate posterior with some optimisation algorithm based on an approximation of the ELBO. Early algorithms for non-conjugate models used tailored approximations of the ELBO combined with a suitable, e.g. natural gradient optimisation algorithm (e.g. Honkela et al., 2010) while more recent methods such as ADVI (Kucukelbir et al., 2016) have focused on stochastic approximation of the ELBO combined with stochastic gradient optimisation.

## 3 Methodology

In this section we introduce the proposed DPVB-SGA algorithm. It is a DPVB algorithm that runs on mini batches of data and is based on stochastic gradient ascent (SGA). The variational parameters  $\xi$  are learned by maximising ELBO with SGA. AdaGrad (Duchi et al., 2011) is used to improve the learning rate.

The algorithm requires several parameters to be set. Sampling frequency  $q$  for subsampling within the data set, total number of iterations  $T$  and clipping threshold  $c_t$  are important design decisions that determine the privacy budget. Clipping the gradients using the threshold  $c_t$  enables evaluation of sensitivity of gradients and is one of the important backbones of the method. The algorithm also needs initial values for variational parameters  $\xi_0$  and a learning rate  $\eta$ .

At each iteration a subset  $U$  of the data set  $\mathcal{D}$  is chosen based on  $q$  and gradient for each data sample is calculated and clipped using  $c_t$ . Then, gradient contributions from all data samples in the mini batch are summed and perturbed with Gaussian noise

$\mathcal{N}(0, 4c_t^2\sigma_\delta^2\mathbf{I})$ . The pseudo-code is presented in Algorithm 1. In the next subsections we describe in detail how privacy design parameters are chosen and privacy budget is calculated.

---

**Algorithm 1: DPVB-SGA**

---

**input** : Data set  $\mathcal{D}$ , sampling probability  $q$ ,  
number of iterations  $T$ , SGA step size  $\eta$ ,  
Clipping threshold  $c_t$  and initial values  
 $\xi_0$ .  
**output**: DP variational parameters

**for**  $t \in [T]$  **do**  
    Pick random sample  $U$  from  $\mathcal{D}$  with sampling  
    probability  $q$ ;  
    Calculate gradient for each  $i \in U$ ;  
    Clip and sum gradients:  
     $\tilde{g}_t(x_i) \leftarrow g_t(x_i) / \max(1, \frac{\|g_t(x_i)\|_2}{c_t})$ ;  
     $\tilde{g}_t \leftarrow \sum_i \tilde{g}_t(x_i)$ ;  
    Add noise:  $\tilde{g}_t \leftarrow \tilde{g}_t + \mathcal{N}(0, 4c_t^2\sigma_\delta^2\mathbf{I})$ ;  
    Update AdaGrad parameter.  $G_t \leftarrow G_{t-1} + \tilde{g}_t^2$ ;  
    Ascent:  $\xi_t \leftarrow \xi_{t-1} + \eta\tilde{g}_t/\sqrt{G_t}$   
Calculate privacy budget

---

### 3.1 Choosing clipping threshold

Clipping threshold is chosen before learning, and does not need to be constant. After clipping  $\|g_t(x_i)\|_2 \leq c_t, \forall i \in U$ . Obviously clipping will effect on learning, but it is necessary to provide privacy. Clipping gradients too much will distract SGA, but on the other hand large clipping threshold will cause large amount of noise to sum of gradients.

### 3.2 SGA sample size

Parameter  $q$  determines how large subsample of the training data we use to for gradient ascent. With small  $q$  values we need larger  $T$ . However small  $q$  values cause more amplifying on privacy. Sample size and  $c_t$  effect mutually on the accuracy. This can be seen by keeping  $\sigma$  parameter constant, without loss of generality lets choose  $\sigma = 1$ , and choosing  $q$  so that only 1 individual data point is sampled in each iteration. Now as norm of added noise will be  $\mathcal{O}(\dim(\xi)c_t)$  and norm of sum gradient  $\leq c_t$ , the released private gradient will be dominated by the noise term. While in our experiments  $q$  was fixed, we could also alter the  $q$  during iteration.

### 3.3 Calculating privacy budget

We have chosen to perturb gradients in each iteration with zero mean multivariate normal noise with covariance matrix  $4c_t^2\sigma_\delta^2\mathbf{I}$ . Parameter  $\sigma_\delta$  in noise level deter-

mines our total  $\epsilon$  and depends on the total  $\delta$  in privacy budget. We can calculate by the total privacy budget by setting  $\sigma = \sqrt{2\ln(1.25/(\delta_{iter}))}/\epsilon_{iter}$ . Clipping will lead  $\ell_2$  sensitivity of gradient sum to be  $2c_t$ , so perturbing each sum with aforementioned noise will lead each iteration to be  $(\epsilon_{iter}, \delta_{iter})$ -DP w.r.t the subset. If sampling probability  $q$  amplifies privacy i.e  $\epsilon_{amp} < \epsilon$ , then also  $\delta$  will be amplified and every iteration and SGA will be  $(\log(1 + q(e^{\epsilon_{iter}} - 1)), q\delta_{iter})$ -DP w.r.t the whole data set. Now if we set  $\delta_{iter} = (\delta_{tot} - \delta')/Tq$ , where  $\delta'$  comes from advanced composition, we can provide  $\delta_{tot}$  as  $\delta$  parameter in total privacy cost. The  $\epsilon$  parameter in our total privacy cost will be

$$\epsilon_{tot} = \sqrt{2T\ln(1/\delta')} \sigma' + T\sigma'(e^{\sigma'} - 1),$$

where  $\delta_{iter}$  is chosen as above and

$$\sigma' = \log\left(1 + q\left(\exp\left(\sqrt{2\ln(1.25/\delta_{iter})}/\sigma\right) - 1\right)\right).$$

If we choose not to keep  $\sigma$  constant during training we cannot use advanced composition. However we can sum up individual privacy costs as was stated in section 2.1.1. This kind of perturbation could be useful because it allows to use different noise levels during training, but in our experiments we have used constant  $\sigma$  and  $c_t$ .

## 4 Example: logistic regression

We apply DPVB-SGA to infer posterior of regression coefficient  $\mathbf{w}$  of logistic regression. Our model is:

$$P(y|\mathbf{x}, \mathbf{w}) = \sigma(y\mathbf{w}^T\mathbf{x})$$

$$p(\mathbf{w}) = N(\mathbf{w}; \mathbf{w}_0, \mathbf{S}_0),$$

where  $\sigma(x) = 1/(1 + \exp(-x))$ . We take no prior on the covariance matrix  $\mathbf{S}_0$  which is fixed to  $\mathbf{S}_0 = \mathbf{I}_d$ . We assume that the approximate posterior  $q(\mathbf{w})$  is multivariate normal with mean  $\mathbf{m}_N$  and covariance  $\mathbf{S}_N$  and we denote these variational parameters with  $\xi$ . The individual ELBOs become

$$\mathcal{B}(\xi; y_i) = -\frac{1}{n}\text{KL}(q(\mathbf{w}; \xi) \| p(\mathbf{w})) \quad (6)$$

$$+ \langle \log p(y_i|\mathbf{w}) \rangle_{q(\mathbf{w}; \xi)}. \quad (7)$$

Expectation in the ELBO is intractable because of the form of our likelihood, so we approximate it using Monte Carlo estimation with

$$\langle \log p(y_i|\mathbf{w}) \rangle_{q(\mathbf{w}; \xi)} \approx \frac{1}{L} \sum_{l=1}^L \log p(y_i|\mathbf{w}_l),$$

where samples  $\mathbf{w}_l$  are drawn from  $q(\mathbf{w}) = N(\mathbf{m}_N, \mathbf{S}_N)$ . In order to draw the Monte Carlo sample, we reparametrise  $\mathbf{w}$  as

$$\mathbf{w} = \mathbf{m}_N + \tilde{\mathbf{S}}_N \boldsymbol{\nu},$$

where

$$\boldsymbol{\nu} \sim N(\mathbf{0}, \mathbf{I})$$

and  $\tilde{\mathbf{S}}_N$  is the Cholesky factor of  $\mathbf{S}_N$  satisfying

$$\tilde{\mathbf{S}}_N \tilde{\mathbf{S}}_N^T = \mathbf{S}_N.$$

Now we can rewrite (6) as

$$\begin{aligned} \mathcal{B}(\boldsymbol{\xi}; y_i) &\approx -\frac{1}{n} \text{KL}(q(\mathbf{w}; \boldsymbol{\xi}) \| p(\mathbf{w})) + \\ &\frac{1}{L} \sum_{l=1}^L \log \sigma(y_i(\mathbf{m}_N + \tilde{\mathbf{S}}_N \boldsymbol{\epsilon}_l) \mathbf{x}_i). \end{aligned}$$

The KL-divergence between two multivariate normal distributions can be written as

$$\begin{aligned} \text{KL}(q(\mathbf{w}; \boldsymbol{\xi}) \| p(\mathbf{w})) &= \frac{1}{2} \left( \ln \frac{|\mathbf{S}_0|}{|\mathbf{S}_N|} + \text{tr}(\mathbf{S}_0^{-1} \mathbf{S}_N) \right) + \\ &\frac{1}{2} [(\mathbf{m}_N - \mathbf{w}_0)^T \mathbf{S}_0^{-1} (\mathbf{m}_N - \mathbf{w}_0)]. \end{aligned}$$

Using the assumption  $p(\mathbf{w}) = N(\mathbf{0}, \mathbf{I})$ , the above expression simplifies to

$$\begin{aligned} \text{KL}(q(\mathbf{w}; \boldsymbol{\xi}) \| p(\mathbf{w})) &= \frac{1}{2} \left( \ln \frac{1}{|\mathbf{S}_N|} + \text{tr}(\mathbf{S}_N) \right) + \frac{1}{2} \mathbf{m}_N^T \mathbf{m}_N \\ &= \frac{1}{2} \left( -\ln |\tilde{\mathbf{S}}_N|^2 + \sum_{k=1}^d \|\tilde{\mathbf{S}}_{N_k}\|^2 \right) + \frac{1}{2} \mathbf{m}_N^T \mathbf{m}_N \\ &= -\frac{1}{2} \ln |\tilde{\mathbf{S}}_N|^2 + \frac{1}{2} \sum_{k=1}^d \|\tilde{\mathbf{S}}_{N_k}\|^2 + \frac{1}{2} \mathbf{m}_N^T \mathbf{m}_N. \end{aligned}$$

## 5 Experiments

We tested logistic regression using the Abalone data set from the UCI Machine Learning Repository (Lichman, 2013) for the binary classification task. Individuals were divided into two classes based on whether individual had less or more than 10 rings. The data set consisted of 4177 samples with 8 attributes. Training of classifier was performed using 80% of the data using stochastic gradient with sampling ratio  $q = 0.02$ . Before training, features of the data set were standardised by subtracting feature mean and dividing by feature standard deviation.

In order to test the scalability of the method to more samples, we also tested it with a synthetic data set with 7 features and 50000 samples from which 80% was used for training the classifier. In addition to that we compare the performance on two subsets of the entire data set with 10000, and 20000 samples. We generated the synthetic data set by drawing each sample of  $\mathbf{x}_i$  from  $\mathcal{N}_d(\mathbf{0}, \mathbf{I})$ . The coefficient  $\mathbf{w}$  was drawn from

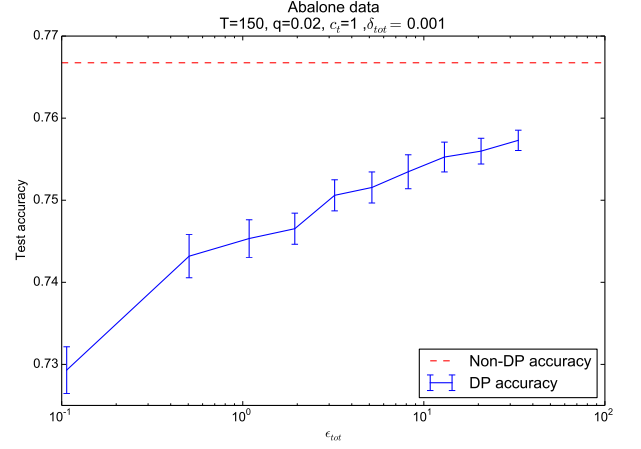


Figure 1: Accuracy vs. total  $\epsilon$  in an example setting with the Abalone data. The curve shows the mean of 10 runs of the DP algorithm with error bars denoting the standard error of the mean.

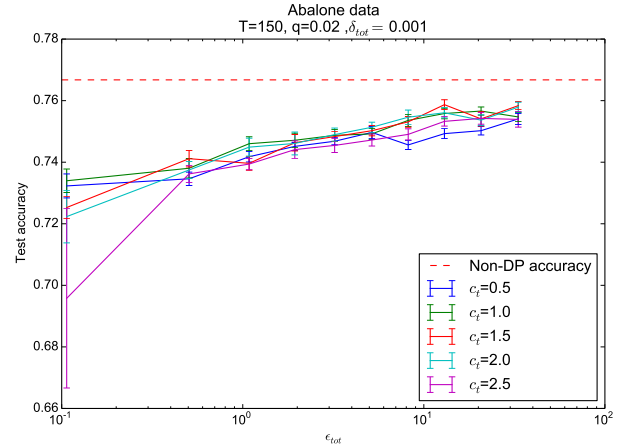


Figure 2: Accuracy vs. total  $\epsilon$  in Abalone data with several clipping threshold values. The curve shows the mean of 10 runs of the DP algorithm with error bars denoting the standard error of the mean.

$\mathcal{N}_d(\mathbf{0}, \mathbf{I})$ . For each  $\mathbf{x}_i$  we calculate the class probability  $\mathbf{z}_i = \sigma(\mathbf{w} \cdot \mathbf{x}_i)$  and draw the target class  $y_i$  according to this probability.

From Figure 1 we can see, that our private classifier performs quite well with relatively small  $\epsilon_{tot}$  values. Compared to the 77% classification accuracy of the non-private algorithm, even with  $\epsilon_{tot} = 0.1$  we still classify 73% of the test set correctly with the accuracy approaching non-private level as  $\epsilon_{tot}$  increases.

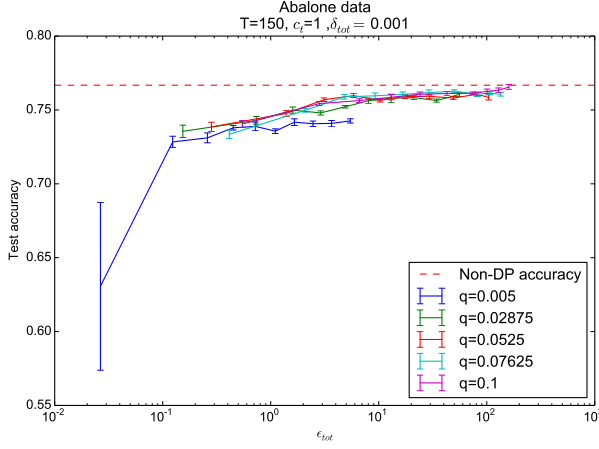


Figure 3: Accuracy vs. total  $\epsilon$  in Abalone data with several SGA sample sizes. The curve shows the mean of 10 runs of the DP algorithm with error bars denoting the standard error of the mean.

### 5.1 The effect of gradient clipping

As was mentioned before, there are many parameters that we can change and Figure 2 shows the effect of gradient clipping threshold. We can see that aggressive clipping with small  $c_t$  values is useful in overcoming the effect of large noise with a tight privacy budget corresponding to a small  $\epsilon$ , but under a looser privacy budget, clipping will start hurting the learning more. Clipping the gradients too little is also bad because the increase in the level of added noise will be more significant than the increase in the retained information because of less clipping.

### 5.2 The effect of subsampling ratio

The effect of SGA sampling ratio  $q$  is shown in Figure 3. The figure shows that  $q = 0.005$  corresponding to a minibatch size of approximately  $q \cdot 0.8 \cdot 4177 \approx 17$  is clearly inferior to the larger values of  $q$ . Presumably the level of noise added is too strong relative to the magnitude of the gradient at this sample size. There are no clear trends in the performances of the other sampling ratios, suggesting that minibatch sizes in the range 96...334 seem to perform reasonably well.

### 5.3 The effect of number of iterations

It is clear that the performance of our classifier depends greatly on the convergence of variational parameters. On the other hand in order to maintain fixed privacy budget we need to add more noise per iteration for longer runs of the algorithm. The effect of iteration number  $T$  on test accuracy is shown in Figure 4. There do not appear to be significant differences

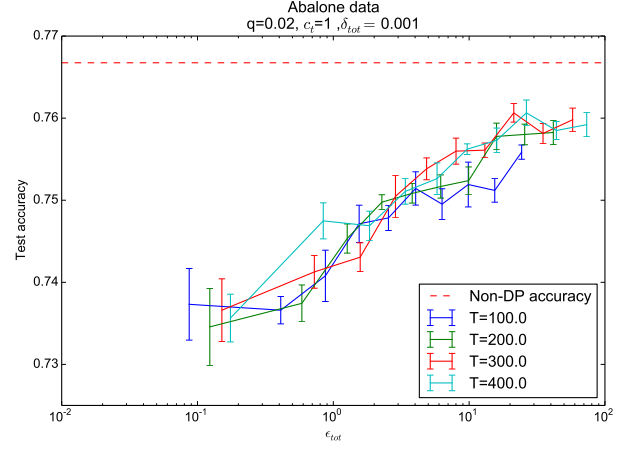


Figure 4: Accuracy vs. total  $\epsilon$  in Abalone data with different number of iterations. The curve shows the mean of 10 runs of the DP algorithm with error bars denoting the standard error of the mean.

between the different values tested here, with the exception of the smallest number  $T = 100$  which seems to perform clearly worse in the high  $\epsilon_{tot}$  regime.

### 5.4 The effect of number of samples

In order to test the method with larger data sets, we also applied it to a synthetic data set with  $n = 50000$  samples also testing smaller subsets. We fixed  $T = 500$  and  $c_t = 2$  for the synthetic data set and compared the accuracies between two different  $q$  values. Results are shown in Figure 5. As the number of samples grows, we can see that classifier both becomes more accurate and approaches the accuracy of the non-private version. This is expected because the number of samples itself doesn't really affect on our noise level, but sampling ratio  $q$  on the other hand does change our noise level. For example between  $n = 50000$  and  $n = 10000$  with  $q = 0.001$ , the number of samples used in each iteration is 5 times bigger with  $n = 50000$  than with  $n = 10000$ , but still our noise level stays the same. We can see that with  $n = 20000$  we are very close to non-private classifiers accuracy even with  $\epsilon_{tot} = 0.1$  and with  $n = 50000$  our private classifier with  $\epsilon_{tot} \geq 0.1$  is practically indistinguishable from the non-private one.

Comparing the different values of  $q$  on the synthetic data we see that  $q = 0.001$  is inferior to  $q = 0.01$  for  $n \leq 20000$  while there does not appear to be a significant difference for  $n = 50000$ . This further confirms that the smaller minibatch sizes of 8 and 16 for  $n = 10000$  and  $n = 20000$ , respectively, are too small, while the minibatch size of  $0.001 \cdot 0.8 \cdot n = 40$  with  $n = 50000$  is already sufficient.

## 6 Discussion

There is currently a strong dichotomy in differentially private Bayesian methods. Conjugate exponential models with a finite sufficient statistic can be handled very efficiently through perturbation of the sufficient statistic using the Laplace mechanism. As shown by Foulds et al. (2016) and Honkela et al. (2016) this approach is both consistent and efficient: as the size of the data set increases the private estimates converge to the corresponding non-private variants at an optimal rate of  $1/n$ .

The situation with non-conjugate models is much more difficult and no similarly efficient algorithms are known. Algorithms such as the posterior sampling of Wang et al. (2015) and our DPVB are iterative and carry a privacy cost for each iteration, which means that the number of iterations basically has to be fixed beforehand if one wishes to adhere to a fixed privacy budget. The algorithms cannot be guaranteed to be run until convergence with a fixed privacy budget. While this is theoretically unpleasant, in practice it can be seen to reflect the situation that some problems are difficult and may not be solvable to a high accuracy under a tight privacy budget.

Differential privacy and high dimensionality do not go well together. This can also be seen in the norm of the noise added to the gradients in our algorithm, which depends linearly on the dimensionality of the variational parameter. This can create another interesting trade-off for higher dimensional data sets: a posterior approximation with a full covariance is in general more accurate (see also Kucukelbir et al., 2016), but as it requires many more parameters at some point the DPVB algorithm with a diagonal covariance is likely to yield more accurate results.

As demonstrated in the experiments, some level of gradient clipping can increase the accuracy of the method as less noise needs to be added to the more tightly bounded gradients. In theory, gradient clipping is a safe operation as it will almost surely not change the fixed points of the gradient algorithm. In practice things may be more complicated as the clipped gradients may affect the convergence rate and under a fixed iteration budget implied by a fixed privacy budget too aggressive clipping may hurt the results.

In the experiments we also tested various levels of subsampling of the data. Based on all the results it seems that a minibatch of size 20 or less will be too small as presumably the noise added to the gradients becomes too dominant. Minibatches of size 50–100 or more seem to work well.

The variational inference framework used in this work

is very similar to the Automatic Differentiation Variational Inference (ADVI; Kucukelbir et al., 2016). It should be relatively straightforward to combine our method and the ADVI framework to develop a differentially private ADVI method, making it very easy to apply the method to arbitrary new models.

## Acknowledgements

This work was funded by the Academy of Finland (Centre of Excellence COIN; and grants 278300, 259440 and 283107).

## References

- K. Chaudhuri and C. Monteleoni. Privacy-preserving logistic regression. In *Adv. Neural Inf. Process. Syst.* 21, pages 289–296, 2008.
- Christos Dimitrakakis, Blaine Nelson, Aikaterini Mitrokotsa, and Benjamin I. P. Rubinstein. Robust and private Bayesian inference. In *ALT 2014*, volume 8776 of *Lecture Notes in Computer Science*, pages 291–305. Springer Science + Business Media, 2014.
- John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159, July 2011. ISSN 1532-4435. URL <http://dl.acm.org/citation.cfm?id=1953048.2021068>.
- Cynthia Dwork. Differential privacy. In *Proc. 33rd Int. Colloq. on Automata, Languages and Prog. (ICALP 2006)*, Part II, pages 1–12, 2006.
- Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, August 2014. ISSN 1551-305X. doi: 10.1561/04000000042. URL <http://dx.doi.org/10.1561/04000000042>.
- James Foulds, Joseph Geumlek, Max Welling, and Kamalika Chaudhuri. On the theory and practice of privacy-preserving Bayesian data analysis. In *Proc. 32nd Conf. on Uncertainty in Artificial Intelligence (UAI 2016)*, 2016.
- Zoubin Ghahramani and Matthew J. Beal. Propagation algorithms for variational Bayesian learning. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 507–513. MIT Press, 2001.
- Antti Honkela, Tapani Raiko, Mikael Kuusela, Matti Törnio, and Juha Karhunen. Approximate Riemannian conjugate gradient learning for fixed-form variational Bayes. *J Mach Learn Res*, 11:3235–3268, Nov 2010.

Antti Honkela, Mrinal Das, Onur Dikmen, and Samuel Kaski. Efficient differentially private learning improves drug sensitivity prediction. 2016. arXiv:1606.02109.

Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. An introduction to variational methods for graphical models. *Mach. Learn.*, 37(2):183–233, November 1999. ISSN 0885-6125. doi: 10.1023/A:1007665907178. URL <http://dx.doi.org/10.1023/A:1007665907178>.

Alp Kucukelbir, Dustin Tran, Rajesh Ranganath, Andrew Gelman, and David M. Blei. Automatic differentiation variational inference. 2016. arXiv:1603.00788.

Ninghui Li, Wahbeh Qardaji, and Dong Su. On sampling, anonymization, and differential privacy or, k-anonymization meets differential privacy. In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '12, pages 32–33, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1648-4. doi: 10.1145/2414456.2414474. URL <http://doi.acm.org/10.1145/2414456.2414474>.

M. Lichman. UCI machine learning repository, 2013. URL <http://archive.ics.uci.edu/ml>.

Yu-Xiang Wang, Stephen E. Fienberg, and Alexander J. Smola. Privacy for free: Posterior sampling and stochastic gradient Monte Carlo. In *Proc. 32nd Int. Conf. Mach. Learn. (ICML 2015)*, pages 2493–2502, 2015.

O. Williams and F. McSherry. Probabilistic inference and differential privacy. In *Adv. Neural Inf. Process. Syst.* 23, 2010.

J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao. PrivBayes: Private data release via Bayesian networks. In *SIGMOD'14*, pages 1423–1434, 2014.

Jun Zhang, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, and Marianne Winslett. Functional mechanism: Regression analysis under differential privacy. *PVLDB*, 5(11):1364–1375, 2012.

Zuhe Zhang, Benjamin Rubinstein, and Christos Dimitrakakis. On the differential privacy of Bayesian inference. In *Proc. Conf. AAAI Artif. Intell.* 2016, 2016.

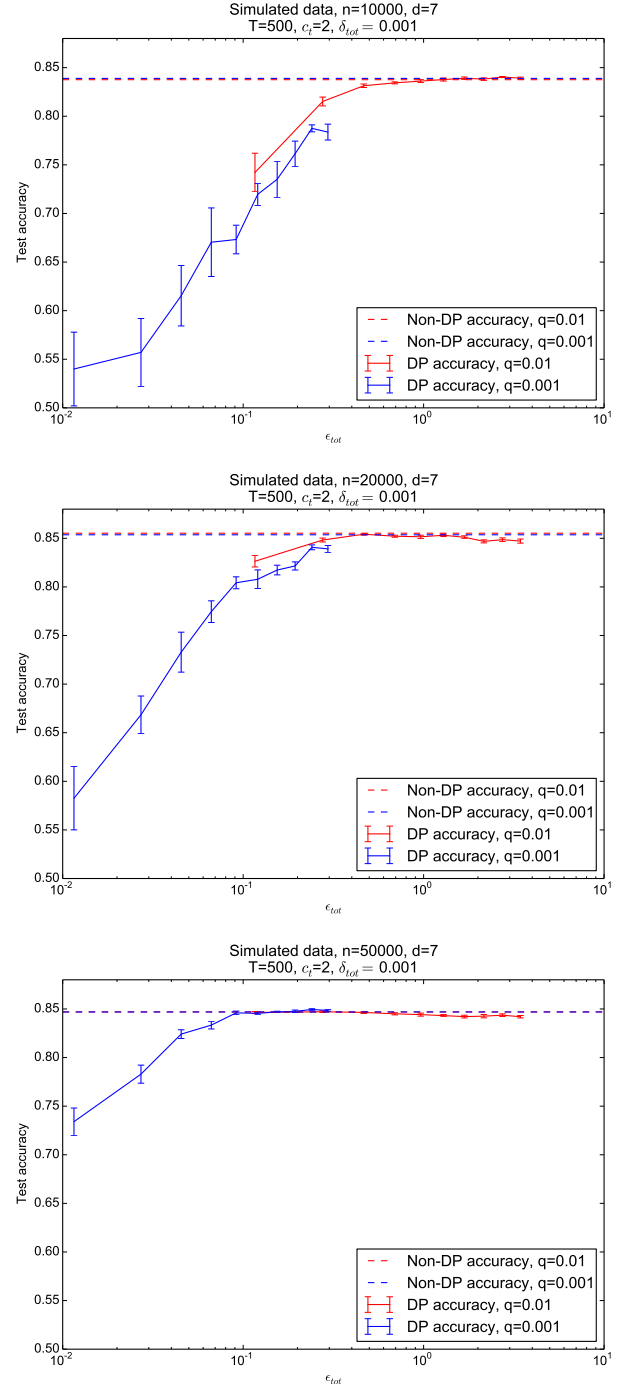


Figure 5: Accuracy vs. total  $\epsilon$  in the synthetic data with different numbers of samples:  $n = 10000$  (top),  $n = 20000$  (middle) and  $n = 50000$  (bottom). The curves show the mean of 10 runs of the DP algorithm with error bars denoting the standard error of the mean.